VMDIS 8003

VME Display

User's Manual, version 2.0
Designation: DOC 8003/UM
PN: 085.169

Version 2.0 - January 1991

CREATIVE ELECTRONIC SYSTEMS S.A.

Creative Electronic Systems

VMDIS 8002/3/3A

USER'S MANUAL

The information in this document has been checked carefully and is thought to be entirely reliable. However, no responsibility is assumed in case of inaccuracies. Furthermore, CES reserves the right to change any of the products described herein to improve reliability, function or design. CES neither assumes any liability arising out of the application or use of any product or circuit described herein nor conveys any licence under its patent rights or the rights of others.

WARNING

THIS EQUIPMENT GENERATES, USES AND CAN RADIATE RADIO FREQUENCY ENERGY AND MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTION MANUAL. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS OF A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCES WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE; IN WHICH CASE, THE USER AT HIS OWN EXPENSE WILL BE REQUIRED TO TAKE WHATEVER MEASURES ARE NECESSARY TO CORRECT THE INTERFERENCE.

Version 2.0 - January 1991

Creative Electronic Systems,
70 Route du Pont-Butin
P.O. Box 107
CH-1213 Petit-Lancy 1, Geneva, Switzerland.

CONTENTS

Chapter 1

INTRODUCTION

1.1 From the euphoria of design to the nightmare of debugging

As long as VME is used in a small development from the same manufacturer there is no need for a VMEbus diagnostic tool as everything should occur as described in the manuals.

When one begins to build VME systems consisting of hardware and software from different sources (i.e a CPU from A, a memory from B, a disk interface from C and software from yet another company), it is very likely that things are not going to occur as you expect them to go.

1.2 A simple way to check your problems

A simple bus display will then help you to check for the basic problems such as system Clock not inserted, short circuits between address or data lines, VME timing signals being correctly generated, etc...

1.3 Display synchronized by the CPU board

A further step will be to be able to synchronize the display on each cycle emitted by your CPU board and to check the validity of data and the sequence in which it is handled.

1.4 Who is the culprit ?

Should you be using a VME system involving advanced hardware such as multiple CPUs and distributed software, you must be ready to face much more complex problems, one of the most frequent of which is: Which CPU board made that nice crash ? Then you obviously need even more elaborate tools than the ones mentioned above.

1.5 A light and compact tool

Finally if you are a field-service engineer using this module as a tool for debugging widely-dispersed systems, you will be happy that it is so light and compact.

Relax and sit back

You have at least bought the right tool if you are reading this manual.

Chapter 2

AVAILABLE RESOURCES

A brief description of the front panel gives an accurate idea of the possibilities you have at hand.
The status of each line of the VME bus is displayed on the front panel by a Led.
Led "On" means line "On".

The lines are grouped as follows, counting from the top:

2.1 Address and Data lines display

The 32 bits of the VMEbus address lines are displayed together with the 32 bits of the data lines

Left side	A01 to A31	Addresses, A01 being the LS Address line
Right side	D00 to D31	Data, D00 being the LS Data line

A00 is used for byte separation in the address and is used in byte mode to drive DS0 and DS1.

If the data is sent with a "MOV Long" instruction, an additional Led labelled "LWD" (Long Word) will be lit. Data can be transmitted in byte mode (MOVB), word (16 bits) mode (M0V), or long word (32 bits) mode (MOVL).

Each set of 4 leds is marked with a vertical bar to show hexadecimal grouping.

Refer to § 2.9 for Address and Data lines display location.

2.2 Address Modifier display

The 6 bits of the address modifier lines from AM0 (LSB) to AM5 (MSB) are displayed on the left side of the front panel.

These lines are extremely important, because they form, with the address lines, the only resource identification from the VMEbus side. This means that a physical resource is characterized by a combination of address and address modifier.

One of the most common problem encountered in VME systems is that the address modifiers issued do not match those decoded by the unit you want to address.

Do verify that the address and address modifier combination emitted on the VMEbus correspond to the configuration of the addressed unit.

Refer to § 2.9 for Address Modifier display location.

2.3 Bus Request and Bus Grant display

The 4 Bus Request levels (BR0 to BR3) and the 4 Bus Grant levels (BG0 to BG3) are displayed on the right side of the unit.

Each time a CPU wants to become Master on the VMEbus, it must emit a Bus Request on a given level (BR0 being the lowest and BR3 the highest). Access to the bus will be granted by the VME arbiter. It is signalled by a Bus Grant response at the same level.

A pending Bus Request indicates an abnormal situation.

Refer to § 2.9 for Bus Request and Bus Grant display location.

2.4 Control Signals display

All VMEbus control signals are displayed, some on the left side and some on the right side, in the following order :

	LED	Signal
Left Side	Read	Non Write Cycle
	LWD	Long Word Cycle
	DS0 and DS1	Data Strobes (byte access)
	DTACK	Data Acknowledge
	BERR	Bus Error
	BBSY	Bus Busy
	ACFL *	AC Fail
	SCLK **	System Clock
	STBY *	5 V Standby
Right Side	AS	Address Strobe
	BCLR *	Bus Clear
	SFAIL *	System Fail
	SRES *	System Reset

Please refer to the VMEbus manual for a detailed description of each of these lines.

* These signals are driven asynchronously with reference to the VME cycles. They are stretched observations.

** SCLK may not necessarily be synchronous with reference to the VME cycles that is when another internal clock is used to drive the cycles.

Refer to § 2.9 for Control Signals display location.

2.5 Interrupt Lines Display

The 7 Interrupt Request Lines labelled IRQ1 to IRQ7 (IRQ1 being the lowest level) are displayed together with the Interrupt Acknowledge Line (IACK line).

Refer to § 2.9 for Interrupt Lines display location.

2.6 Display operation Leds and Control setting

The VMDIS 8003 can be operated in 3 different modes, each mode corresponding to a given step in the problem tracing operation.

Choice between these modes is controlled by a front panel toggle switch which each time pushed to the right changes the display operation on a round-robin basis in the sequence:

LATCH - DIRECT - MONITOR.

Each mode is displayed by a corresponding Led on the left hand side.

Operation in these 3 modes is explained in chapter 3, 4 and 5.

These Leds are NOT connected to VMEbus lines but display the internal setting of the unit.

So much for the Leds.

Below the Led section is a mini switch section which controls the optional triggered operation.

For the moment, just make sure that the trigger is wide open, that means that NO trigger condition is selected. To bypass all trigger conditions, *push* all mini switches down on the *left* side.

Each condition is disabled if pushed down on the left, enabled if pushed down on the right.

Refer to § 2.9 for Operation Leds display and Control setting location.

2.7 Front panel toggle switch

This switch controls 2 functions:

- when pushed Left : it issues a display reset command
- when pushed Right : it controls the display operating mode.

Display reset

A display reset is MANDATORY when you power-on the system. It clears all Leds and sets the display in the normal operation mode (LATCHED MODE).

The control of the display operation mode has been described in the preceding paragraph. (refer to § 2.6)

2.8 External trigger input

On the bottom of the unit is a Lemo 00 connector which allows the display to be gated on an external signal.

This condition can be the ONLY trigger condition, or it can be added to the internal trigger condition. It accepts TTL levels - positive logic +5 V = "ON".

A common application is to connect it to the internal Bus Busy signal (BBSY) from a VME CPU board in the same card cage, or VME Control (VMECTL) signal available on the front panel of CES VME Master units (SBC 8221, SBC 8230, VBR 8212, etc ...).

With such a connection the VMDIS will only display the VME cycles issued by that particular VME Master.

Refer to § 2.9 for External trigger input location.

2.9 Front panel

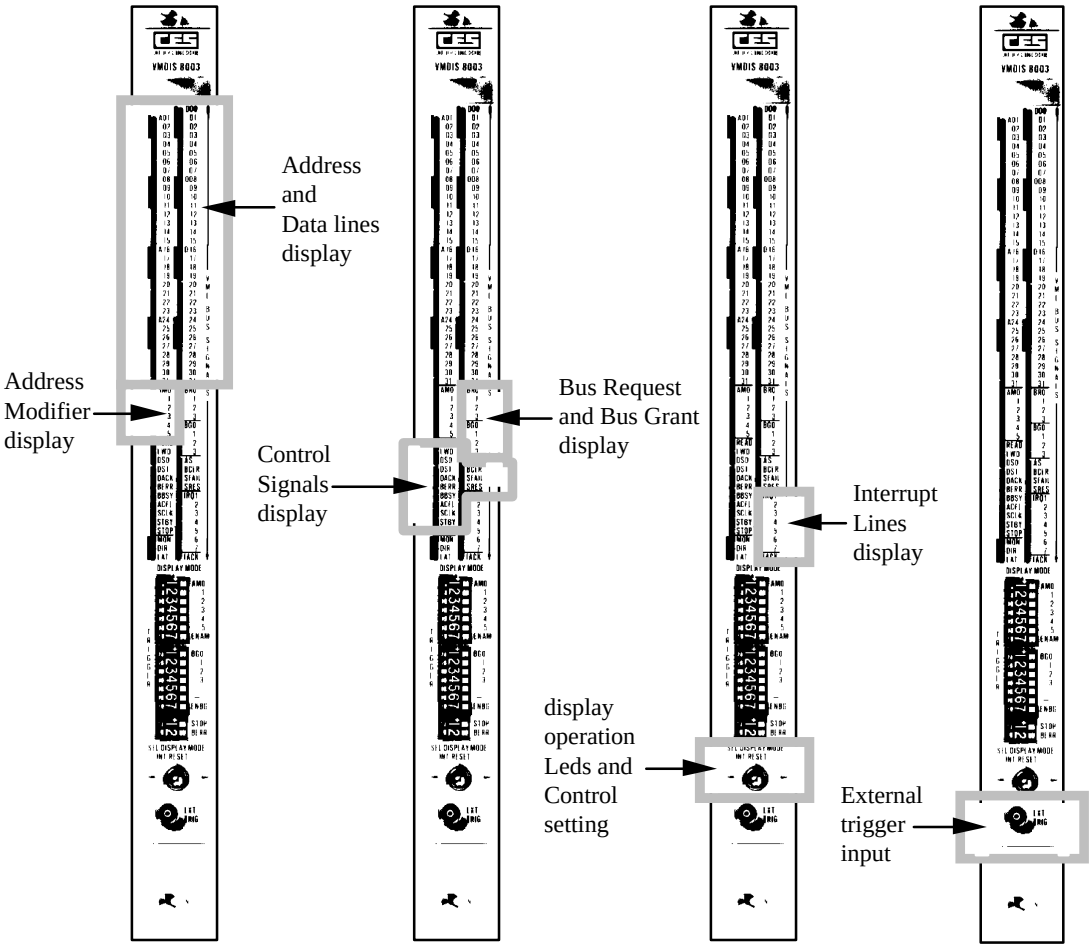


Fig 2.1 : Front panel

Chapter 3

NORMAL OPERATING MODE

The normal operating mode is the Latched mode.

3.1 Introduction

In this mode, all the VME transactions in the card-cage are displayed on the front panel, even if they do not address the VMDIS itself. The VME signals are latched into the VMDIS registers. The stretched lines (SYSFAIL, SYSRESET, BCLR, BR0, BR1, BR2, BR3) are asynchronous. If trigger conditions are enabled, only the cycles meeting the trigger pattern will be memorized. The Address Strobe Led (AS) will be lit only for such cycles.

N.B.: When using this mode, the trigger conditions are of primordial importance, so start with an "open" trigger and then refine the trigger step by step in an iterative way.

3.2 VME read-out

Moreover the VME signals may be latched into the VMDIS internal registers and thus they can be retrieved via VME for further examination.

To do this, a jumper labelled "GEO" must be inserted; then the information is available by READ commands with Address Modifiers 01 at the following addresses:

Address	MSB Contents LSB	Comments
...000	D31 ---------- - D16	Data HW
...001	D15 ---------- - D00	Data LW
...010	A31 ---------- - A16	Address HW
...011	A15 ---------- A01, LWord	Address LW
...100	X, X, X, BR3, BR2, BR1, BR0, ACFAIL BG3, BG2, BG1, BG0, READ, IRQ1, IRQ2, IRQ3	Status HW
...101	IRQ4, IRQ5, IRQ6, IRQ7, BERR, DTACK, DS0, DS1 BBSY, IACK, AM05, AM04, AM03, AM02, AM01, AM00	Status LW

Signal timing in "LAT" mode:

The VME signals are stored as follows:

1) A01-A31, LWORD, IACK: on the leading edge of AS*

2) D00-D31, DS0, DS1, WRITE*, BBSY*: on the leading edge of DTACK or BERR

3) DTACK, BERR: as they occur

NB.: '*' means active low signal.

Once the display operation in this mode has been understood, the trigger facilities can be used.

3.3 Trigger operation

The trigger logic allows triggering on two fields:

- the AMx lines
- the BGx lines

The Bus Grant selection accepts either daisy-chain priority or distributed priority schemes according to jumper setting. According to its position in the crate, the VMDIS will display the corresponding priority; more than one VMDIS 800x can be used per crate to identify the priority of each processor.

WARNING: Only one VMDIS in the card-cage may have the jumper "GEO" inserted.

N.B. The VMDIS always stores and displays the last arbitration cycle.

Each field can be enabled or disabled by a mini switch on the front panel. The field is enabled when the switch is depressed on the side of the label (i.e ENAM and ENBG). If the field is enabled, the trigger conditions for this field must be selected by the corresponding mini switches, which are "ON" when depressed on the side of the label.

Example 1 :

VMDIS 8003 should only display cycles having AMx = 101100 and any BG. The following set-up is made:

ENAM "ON " with AM5 "ON" ENBG "OFF"
AM4 "OFF"
AM3 "ON"
AM2 "ON"
AM1 "OFF"
AM0 "OFF"

Example 2 :

VMDIS 8003 should display all cycles on BG level 2.

ENAM "OFF" ENBG "ON"
BG0 "OFF"
BG1 "OFF"
BG2 "ON"
BG3 "OFF"

WARNING : only one BGx bit may be "ON".

3.3.1 After trigger operation

Once the trigger conditions have been meet, two different modes of operation can be selected:

- either FREE RUN
- or STOP

In FREE RUN mode, the VMDIS 8003 will memorize and display continuously the cycles meeting the trigger conditions.

The STOP mode is selected by depressing the corresponding mini switch on the side of the label. In STOP mode, the VMDIS 8003 will memorize and display only the first cycle satisfying the trigger conditions and will then enter an internal Stop state. This state will be indicated by the STOP Led flashing. All further transactions are ignored. To return to normal mode, the INT. RESET switch must be pushed.

3.3.2 External trace operation

An external condition can be added to the internal trigger conditions by feeding a signal to the front panel Ext. TRIG. LEMO 00. The convention is positive logic. This external condition is "ANDed" with the internal conditions.

3.4 Stop on Bus ERROR

An additional facility of the VMDIS 8003 is the possibility to stop the display on a cycle containing BERR. This facility is enabled by depressing the corresponding mini switch on the side of the label.

BEWARE stop-on-BERR overrides all other trigger conditions.

3.5 Replies to DTACK and BERR

VMDIS 800X answers DTACK in all cycles containing AMx = 000000, and in all correct transactions containing AMx = 000001.

VMDIS 800X answers BERR in the following conditions:

- read back cycle with DS0 * DS1 = 0
- read back cycle with LWORD and A01 = 1.

In all other cases the VMDIS ignores the VME handshake cycles.

Chapter 4

CONTROLLED OPERATING MODE

The computer controlled operating mode is the Monitor Mode.

4.1 Introduction

In this mode the VMDIS will not display all transactions occurring on the VMEbus, but only those issued to its own address.

The role of this mode is to perform a computer-controlled test of some features of the VME system, with a CPU writing data into the VMDIS itself and reading it back to check for discrepancies.

This mode is of prime importance for large VME systems distributed over long distances where it is difficult to check the operation visually.

For example, VME systems using the CES VMVbus covering more than 100 meters might be subject to a computer driven check before a manual check.

In this mode the VMDIS can be used to display a message to the operator, such as results of tests, actual phase within the process control, etc...

4.2 Operation

In this mode, all the VME signals are memorized in the VMDIS 800X if they are sent with AMx = 000000. The data relevant to these signals can be read back with AMx = 000001, both in LONGWORD and WORD modes.

Note that the standard address modifier decoding is AMx = 000000, 000001. This may be changed in a fuse-prom.

WARNING: Modifying the AM codes is not recommended.

When running this mode, verify that ALL trigger conditions including Stop and BERR are disabled.

4.3 The effect of "GEO" jumper

The jumper labelled "GEO" has been introduced so that multiple VMDIS modules may be used in a single crate.

When this jumper is installed :

a) the module can be switched between the modes LAT - MON and vice-versa by software, using WRITE cycle, AM = 01
   if AD01 = 1, switch to LAT
   if AD01 = 0, switch to MON
   (in both cases the module replies with DTACK)

b) the data, address and status of the last cycle, stored by VMDIS in LAT mode, can be read by READ AM01 at the addresses defined in § 3.2.

4.4 Replies to DTACK and BERR

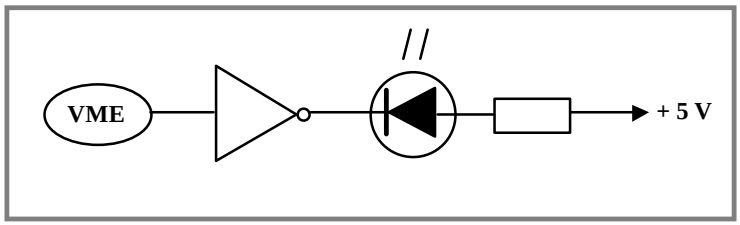
Same as § 3.5.

Chapter 5

TRANSPARENT OPERATING MODE

The transparent operating mode is the Direct Mode.

In this mode, the VMDIS 800X displays the state of the VME bus lines as driven by the VMEbus. This mode can be represented by the following scheme:



For a module equipped with the Trigger option, the Led (AS) indicates cycles which meet the trigger pattern.

In this mode you can check that no VME line is driven permanently by another module.

WARNING: in VME standard some lines are in positive logic, where "High" is Led "ON", other are in negative logic where "High" is Led "OFF".

For example SCLK can be checked in this mode.

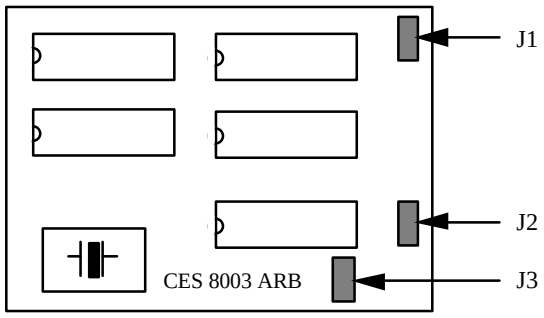
Chapter 6

SLOT 1 FUNCTION OPTION (8003A)

The VMDIS can be equipped with a piggy back board implementing the following Slot 1 functions:

- system clock 16 MHz with ON / OFF control by jumper.
- arbitration on the 4 levels in mode PRI with BCLR generation with ON / OFF selection by jumper.

When used in Slot 1, it is possible to effect the connection IACK IN - IACK OUT via an internal jumper in case no connection can be made through the backplane.



J1 : The Arbitration function is implemented when this jumper is plugged into the board.
J2 : ON / OFF System clock 16 MHz control jumper.
J3 : ON / OFF Bus Clear (BCLR) generation jumper.

WARNING: If J1 jumper is not plugged into the board, so must be J3 jumper.

Appendixes

A.1 Led States in DIR - LAT - MON Modes

The following table resumes the Leds states in the three different modes :

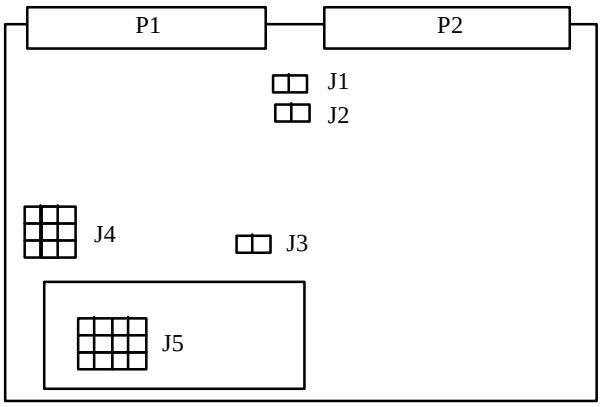
LED	Direct Mode	Latched Mode	Monitor Mode
Ax, Dx, AMx	T	L	L
LWORD	T	L	L
READ	T	L	L
DS0, DS1	T	L	L
DTACK, BERR	T	L	L
BBSY	T	L	L
STBY	D	D	D
ACFAIL	T	T/L	T/L
SYSCLK	D	D	D
BRx	T	S	S
BGx	M	M	M
AS	Ts	Ts	Ts
BCLR	T	S	S
SYSFAIL	T	S	S
SYSRESET	T	S	S
IRQx	T	T/L	T/L
IACK	T	L	L

with :

T	:	transparent
L	:	latched
D	:	direct (not multiplexed)
T/L	:	transparent or latched according to jumper position
S	:	stretched asynchronous
M	:	memorized in the last arbitration cycle
Ss	:	stretched if OK with trigger.
Ts	:	transparent if OK with trigger

N.B. when in STOP mode, the stretched signals BRx and BCLR as well as the direct signals are still active.

A.2 Jumpers position and description



Jumper J1 for daisy-chaining the IACK when the VMDIS is in slot one.

 : IACK chained.

 : Chain broken.

Jumper J2 "GEO" : refer to § 4.3 for explanations.

 : "GEO" enabled.

 : "GEO" disabled.

Jumper J3 : Control display mode of the lines: IRQx, READ, B4.

 : Saved with driver cycle.

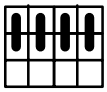
 : Transparent

Jumper J4 : Trigger arbitration mode.

 : VME (standard) arbitration.

 : CERN distributed arbitration.

Jumper J5 : Display mode for LEDs, BR0, BR1, BR2.

 : VME (standard) arbitration.

Jumper settings when in Slot 1

J1 : on
J2 : on or off
J3 : on or off
J4 : VME standard
J5 : VME standard

Jumper settings when not in Slot 1

J1 : off
J2 : on or off
J3 : on or off
J4 : VME standard
J5 : VME standard

A.3 Software example

```
                    1.              TTL     Software examples showing access to VMDIS 8003
                    2.      *               ------------------------------------------------------------------------
                    3.              PLEN 66,6,0
                    4.      *
                    5.      *       Extracted from the file of VMV software examples.
                    6.      *
                    7.      *       Revision :      1.10 of November 1986
                    8.      *       File :          vmdisex.a68
                    9.      *
                    10.     *       The hardware configuration required to run these examples is:
                    11.     *       ------------------------------------------------------------------------------
                    12.     *
                    13.     *       Chassis 1 :     pos 1 : VBR 8212
                    14.     *                               - chassis number set to 5
                    15.     *                               pos 2 : VBE 8213
                    16.     *                               - and do not forget the ribbon cable
          17.       *                       between the P2 connectors of slots 1 & 2.
                    18.     *                               pos 3 : a VME external memory unit.
                    19.     *
                    20.     *       Chassis 2 :     pos 1 : VMDIS 8003
                    21.     *                               - with jumper 'GEO' installed
                    22.     *                               pos 2 : VBR 8212
                    23.     *                               - chassis number set to A
                    24.     *
                    25.     *                               pos 3 : a VME external memory unit.
                    26.     *
                    27.     *       The examples have been all written to use the standard factory
                    28.     *       mapping of the different "spaces"
                    29.     *
                    30.     *               - CSR space: $DFFE00 - $DFFFFF with AM = 3E & 39
                    31.     *               - Window: $E00000 - $EFFFFF with AM = 3E & 39
                    32.     *               - Autorouting A24: $E00000 - $EFFFFF with AM = 3E & 39
                    33.     *
                    34.     *       Should your units not have the standard CES mapping, then you
                    35.     *       must modify the equivalence tables which follow to reflect
                    36      *       the new address
                    37.     *
                    38.     *
                    39.     *       In the following examples the exit conditions are:
                    40.     *               trap #5    : normal end
                    41.     *               trap #6    : error
                    42.     *
                    43.     *
                    44.     *       CSR space (Factory mapping)
                    45.     *       ---------------------------------------
                    46.     *
                    47.     *       These equivalences define the internal registers of the VBE 8213.
                    48.     *
                    49.     *
                    50.     *       1 st part : VBE 8213 Internal Registers.
                    51.     *
# 00DFFE00          52.     csre    equ     $DFFE00;        Control and Status register
# 00DFFE02          53.     inth    equ     $DFFE02;        Interrupt handling register
# 00DFFE04          54.     mapad   equ     $DFFE04;        MAPAD register
# 00DFFE06          55.     mapam   equ     $DFFE06;        MAPAM register
# 00DFFE08          56.     mapcr   equ     $DFFE08;        MAPCR register
# 00DFFE04          57.     vreset  equ     $DFFE04;        Vertical Reset
# 00DFFE0E          58.     timout  equ     $DFFE0E         ;       Time out VMV
                    59.     *
# 00E00000          60.     wind    equ     $E00000 ;       Window start address (standard
                  *                                       version)
                    61.     *
00000000            62.             org     $0
                    63.     *
                    64.     *       example
                    65.     *       -----------
                    66.     *
                    67.     *       Demonstrate write/read/test in a VMDIS 8003 placed in an external
                    68.     *       card cage, and accessed over the Vertical Bus (VMV) in Window Mode.
```

```
                                    69.     *       The VMDIS 8003 must be in Latched mode, and the GEO jumper in
                                *   place.
                                    70.     *
                                    71.     *       initialization of MC68000 internal registers
                                    72.     *       A0      : address in the window
                                    73.     *       A1      : address of data register bits 0-15 within VMDIS
                                    74.     *       A2      : address of address register 0-15 within VMDIS
                                    75.     *       D0      : data written to VMDIS 8003 in chassis #A
                                    76.     *       D1      : bits 0-15 of data register, read back from VMDIS 8003
                                    77.     *       D2      : bits 0-15 of address register, read back from VMDIS 8003
                                    78.     *       D3,D4   : time for address calculation
                                    79.     *       D5      : also the address in the window, for compare operation
                                    80.     *
                                    81.     *
                                    82.     *                               ;       Initialize VBE/VBR
00000000  33FC0100 00DFFE00         83.     deb61:  move.w  #$100,(csre)    ;       Select window mode, mapped AM
00000008  33FC0000 00DFFE02         84.             move.w  #$0,(inth)      ;       no interrupt
00000010  33FC0000 00DFFE04         85.             move.w  #$0,(mapad)     ;       high addr. bits in dest chassis = 0
00000018  33FC0000 00DFFE06         86.             move.w  #$0,(mapam)     ;       Address modifier = 0
00000020  33FC000A 00DFFE08         87.             move.w  #$A,(mapcr)     ;       chassis attached = #$A
00000028  33FC0027 00DFFE0E         88.             move.w  #$27,(timout)   ;       time out VBE set at 4 µs
                                    89.     *
00000030  7000                      90.     deb62:  move.l  #$0,d0          ;
00000032  263C 00E00002             91.             move.l  #$E00002,d3     ;       this address will be window-
                                    92.     *                                       mapped to address 000002 in
                                            *                                       chassis #A thus giving access
00000038  2243                      93.             move.l  d3,a1           ;       to VMDIS data register 0-15
0000003A  263C 00E00006             94.             move.l  #$E00006,d3     ;       and this one to 000006
00000040  2443                      95.             move.l  d3,a2           ;       for VMDIS address register 0-15
                                    96.     *
                                    97.     *               read/write/test loop
                                    98.     *
00000042  2800                      99.     deb63:  move.l  d0,d4           ;       Generate the offset within the
                                            *                                       window by complementing the
00000044  4644                      100.            not.w   d4              ;       data, then making it a word
00000046  2A3C 00E00000             101.            move.l  #$E00000,d5     ;       (16 bits) address in range (FFFE-0)
0000004C  DA84                      102.            add.l   d4,d5           ;       and add the window base.
0000004E  0285 00E0FFFE             103.            andi.l  #$E0FFFE,d5     ;
00000054  2045                      104.            move.l  d5,a0           ;       after all this, d5=a0=$E000xxxx
                                    105.    *                                       write d0 bits 0-15 to the address
                                            *                                       in chassis #A
00000056  33FC 0000                 106.            move.w  #$0,(mapam)     ;       AM=0 will be issued in dest   00DFFE06
                                *                           chassis (#A), thus writing
0000005E  3080                      107.            move.w  d0,(a0)         ;       d0 bits 0-15 into the VMDIS
                                    108.    *                                       read back both data 0-15 and
                                *                           address 0-15 from the registers of
                                    109.    *                   VMDIS 8003.
00000060  33FC 0001                 110.            move.w  #$1,(mapam)     ;       AM=1 will be issued in dest   00DFFE06
                                    *                           chassis
00000068  3211                      111.            move.w  (a1),d1         ;       data bits 0-15
0000006A  3412                      112.            move.w  (a2),d2         ;       address bits 0-15
                                    113.    *                                       compare what we wrote into the
                                114.    *                                   chassis #A with what we read back
                                *                           from the VMDIS.
0000006C  B240                      115.            cmp.w   d0,d1           ;       compare data 0-15
0000006E  6600 0014                 116.            bne     deb64           ;       on error
00000072  BA42                      117.            cmp.w   d2,d5           ;       compare address 0-15
00000074  6600 000E                 118.            bne     deb64
```

```
                        119.    *       increment control parameters
00000078    0640 0001   120.            addi.w  #$1,d0          ;       increment data and repeat until we
0000007C    0C40 0000   121             cmpi.w  #$0,d0          ;       have done FFFF cycles
                        *                                               (d0(0-15)=0)
00000080    66C0        122.            bne     deb63           ;
00000082    60AC        123.            bra     deb62           ;       then start again
                        124.    *       error routine === return to TUTOR
00000084    4E45        125.    deb64:  trap    #5
                        126.    *
                        127.    *
00000086                128.            end
```

A.4 VMEbus backplane connections

Pin Number	ROW a Signal Mnemonic	ROW b Signal Mnemonic	ROW c Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK	A17
22	IACKOUT*	SERDAT*	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12 V	+ 5 V STDBY	+ 12 V
32	+ 5 V	+ 5 V	+ 5 V

J1 / P1 pin assignments

"*" means active low signal.

Pin Number	ROW a Signal Mnemonic	ROW b Signal Mnemonic	ROW c Signal Mnemonic
1	User I/O	+ 5 V	User I/O
2	User I/O	GND	User I/O
3	User I/O	RESERVED	User I/O
4	User I/O	A24	User I/O
5	User I/O	A25	User I/O
6	User I/O	A26	User I/O
7	User I/O	A27	User I/O
8	User I/O	A28	User I/O
9	User I/O	A29	User I/O
10	User I/O	A30	User I/O
11	User I/O	A31	User I/O
12	User I/O	GND	User I/O
13	User I/O	+ 5 V	User I/O
14	User I/O	D16	User I/O
15	User I/O	D17	User I/O
16	User I/O	D18	User I/O
17	User I/O	D19	User I/O
18	User I/O	D20	User I/O
19	User I/O	D21	User I/O
20	User I/O	D22	User I/O
21	User I/O	D23	User I/O
22	User I/O	GND	User I/O
23	User I/O	D24	User I/O
24	User I/O	D25	User I/O
25	User I/O	D26	User I/O
26	User I/O	D27	User I/O
27	User I/O	D28	User I/O
28	User I/O	D29	User I/O
29	User I/O	D30	User I/O
30	User I/O	D31	User I/O
31	User I/O	GND	User I/O
32	User I/O	+ 5 V	User I/O

J2 / P2 pin assignments